



11-2019

Scratch Coding for Adults: Creating a Collectible Game

Karlene T. Clark

University of North Dakota, karlene.clark@library.und.edu

Follow this and additional works at: <https://commons.und.edu/cfl-lp>



Part of the [Library and Information Science Commons](#)

Recommended Citation

Clark, Karlene T., "Scratch Coding for Adults: Creating a Collectible Game" (2019). *Librarian Publications*. 16.

<https://commons.und.edu/cfl-lp/16>

This Book Chapter is brought to you for free and open access by the Chester Fritz Library at UND Scholarly Commons. It has been accepted for inclusion in Librarian Publications by an authorized administrator of UND Scholarly Commons. For more information, please contact zeineb.yousif@library.und.edu.

Scratch Coding for Adults: Creating a Collectible Game

Karlene Tura Clark

Coordinator of Circulation Services & Student Employment

Chester Fritz Library, University of North Dakota

Project Description: *It's Going Swimmily*

Many adults have apprehension regarding coding. Scratch, originally developed by the MIT Media Lab in 2007 for ages 8 to 16, encourages digital citizenship and fundamental computational concepts and practices. Scratch has aided adults in transitioning to more traditional text-based programming languages after gaining the foundational understanding. Scratch is open-source shared software, designed to encourage remixing of projects. No real names are used in account name creation.

This project creates a game with a cartoon octopus that moves, collects fish and avoids a crab moving around the screen. Coders are also taught how to add points and make the animals talk.

As an instructor, a “classroom” can be requested to help manage the projects of a group or across multiple “classrooms” at <https://scratch.mit.edu/educators#teacher-accounts>. The instructor uses their own email for all student accounts in the group. Have the adults get their information to you before the first day of class so they can be added and see each other’s work for inspiration.

Age Range

- Tweens (Ages 8-12)
 - Recommendation for this group is to have at least one other instructor available to circulate, and to only attempt it after foundational projects like *Dinosaurs at the Movies*
- Young Adults (Ages 13-18)
- Adults

Type of Library Best Suited For

- Public Libraries
- Academic Libraries
- School Libraries

Cost

\$0.00

- Libraries already have the necessary equipment, as listed under software / hardware.
The only cost is librarian time in preparation and teaching.

Overview

The use of Sensing (touch reaction), Control (conditional statements), Operators (>. <, =, etc.), and Data (variables) blocks in addition to the foundational blocks are used to create a point collection game. Most of the blocks are used in the creation of a little octopus that collects fish while avoiding a crab. The octopus is controlled by the player and the others are set to move in a pattern across the screen. Both a “you win” and “game over” screen are created.

Software / Hardware Needed

- Computer with speakers
- A network connection
 - The author has found that the Chrome Internet browser works the best
- Method of projecting the instructor project during the lesson
- Current version of Adobe Flash player

Step-by-Step Instructions

Prep

- Create a classroom account ahead of time, if desired
- Create usernames at <https://scratch.mit.edu>
- Have them log in to their accounts.

Engage

- Show them <https://scratch.mit.edu/projects/154781453/> and <https://scratch.mit.edu/projects/155828528/> . Both are creative collectible games created by some of the author's former students, shared with their permission.
 - This demonstrates a few of the many ways collectible games can be created.

Create

- The instructor demonstrates on a screen, letting them follow along with the steps to create a mid-level project: collecting for points.
- Start by clicking on “create” in the upper left corner of the screen, next to the Scratch logo. The coding screen (currently empty) will appear on the right, with coding blocks in the center and their first sprite, Scratch Cat, in the center of the left window.
- At the bottom of the screen, the sprite is outlined in blue. Right click and delete the cat.
- To choose a new one, click the little figure next to New Sprite. Some of them will have only the one image and some will have multiple costumes. For this project, choose Octopus
 - Click on the Costumes tab at the center top. You'll see she has two different costumes. Let's use her!
- She's going to need a backdrop. At the bottom left, click on Stage, and notice the blue box around it – this means it is the active item.
 - New Backdrops can be added at the top center or under the Stage.
 - Just under the words “new backdrop” are images of mountains, a pen, a folder, and a camera. For now, use the mountain portrait to choose a backdrop from the library.
 - Choose “underwater1.”
- Drag the octopus to the upper left corner. Her sprite box should now be outlined in blue again so we can work with her.
- Click on the Scripts tab again.

- To make anything happen, we need an Event (brown) to get things started. Specifically, choose “when green flag clicked.”
- Motion (blue) allows movement, sets locations and determines graph coordinates. To make the octopus always start where she was moved to, now choose the block “go to x: 0 y: 0.” The numbers in the coordinates will be different for each person, depending on where the octopus is resting.
 - Drag the block over to the right pane and attach it under the Flag.
 - We also want her to always be there at the start of the game, so go to the purple Looks blocks and find “show.”
 - Place that under the “go to” block.
- Since we want her to have room to move around the screen, we need to make her smaller.
 - Above the green flag and red stop sign, there are editing tools.
 - Choose the one with the arrows pointed inwards, then click the sprite as many times as you like to make her small enough. This can be further adjusted later. But for now, she has space to move!
- We also want her to be able to move not only left and right, but up and down.
 - Go the Control (yellow) tab and drag over a “forever block” under your blue “go to” block.
 - Under the same tab, you will need four of the “if... then” blocks.
 - Place these one under the other, inside the forever block.
- Next we’ll need to add blocks that tell us what happens in the “if...then” conditions. That happens under Sensing (light blue).
 - There are a number of fun things that can happen in this area. We’ll come back to some of them later, but for now, we need to move the “key space pressed?” into each of the empty hexagons between the “if” and “then.”
 - Notice they have a dropdown? In order, go down the four and change “space” to right arrow, left arrow, up arrow, and down arrow.
 - This is telling our game what keys on the keyboard will control her movements.

- But we still have no answer to “then.” To provide motion... back to the blue Motion tab!
 - You want to drag a “change x by 10” under the first two “if...then” sections.
 - Try starting your game by pressing the green flag above the play screen and move her to the right. 10 steps moves very quickly!
 - Try changing the number from 10 to something smaller, like 8. Try the game again. Much better! However, she’s still only going one direction.
 - In the “left arrow” section, change x to “-8.” Try the game again. She should now be going back and forth.
- Back in school we had to learn graphing and coordinates. Here’s where it put it to use. X is your horizontal line and Y is your vertical line. In other words, X is your back and forth; Y is your up and down. For your last two sections
 - Drag over the “change y by 10” blocks.
 - Do as you did for the X coordinates; change it to 8 and -8.
 - When you press the start flag, she should now move in all four directions.
- Now, she had a second costume.
 - Go to the Looks tab (light purple) and place a “next costume” block under each of the change blocks. She can now swim!
 - However, it’s extremely fast. We want to “control” that, right? To the yellow blocks!
 - “Wait 1 secs” will allow you to slow down how fast she’s moving.
 - A full second is too slow; try .25 or .20.

Congratulations! You have the first part of the project complete! The next step is to create sprites to collect and sprites to create trouble.

- Go back to New sprite and find Fish2.
 - Remember to shrink it down, smaller than the octopus!
 - Add a brown Event for “when the flag is clicked”

- Add a blue Motion for “go to x: 0 and y: 0” (after placing the fish where you would like it).
- Add a purple Looks block to “show.”
- Then we want our fish to move. That involves yellow Control again. Specifically, a “forever” loop.
- Inside that, we need our blue Motion again.
 - This time we want
 - “move 10 steps” (change it to 3)
 - “if on edge, bounce.”
- Press the green flag and see what happens!
- Let’s get another sprite. This time, Fish3.
 - Shrink it down, pick a spot on the screen (not close to the octopus), and... (Pause and see if they remember at this point. They should be getting used to the concept of needing “when the flag is clicked” and “go to.”)
 - Don’t forget a purple Look to “show” him!
 - We’ll need another “forever” loop from... (pause to let them fill in yellow Control).
 - Blue motion and get a “move 10 steps.” Experiment on speed.
 - Now another Control block.
 - “If...then” allows us to set parameters. Put this inside the “forever” loop.
 - Just like the other fish, we need a light blue Sensing block now.
 - This time, we want the fish to leave the screen and reappear on the other side. To do this, we need
 - “touching mouse-pointer” and change the drop down to “edge.”
 - Now move your fish to the far left side of the screen. Get him as close to the edge as you can without his tail fin touching it.
 - Add another “go to x:0 y:0.”

- Change your Y to match the Y in your first instance under your Event block. This will allow your fish to stay in a straight line.
- Try pressing the green flag again. You should now have one fish bouncing back and forth, and one scrolling across the screen.
- Press the stop button to halt the movement. W

We're going to add one more sprite – the enemy. If the octopus touches this one, the game is over.

- Go back to the New sprites and choose Crab.
 - Place him at the bottom of the screen
 - Add your Event and Motion blocks to set his location
 - Add a Look block to keep him visible, and shrink his size.
 - We're going to Control his movements with a Forever loop. Randomly move the crab to a new location
 - Drag a “glide 1 secs to x:0 y:0.”
 - Repeat this six or eight times, ending up near the place he started.
 - When you press the green flag, he should now be moving in the pattern you set for him.

Break Time

This is a natural spot to stretch, look away from the screen, or break for the day, depending on the frustration level of participants or the amount of time it took to create. At this point, they can see that something is happening and that they have created the first level of coding: a moving cartoon. If the adults are comfortable, continue. If not, they can pick up here again the next day.

Back to Coding

If starting over on a new day, review the blocks used the previous day and how they worked.

Now we're going to add orange Data. This is going to involve bouncing back and forth quite a bit between the blocks and the sprites. It's not that bad though!

- Click on “Make a Variable.”
 - Name the variable “Points” and leave it set “for all sprites.”
 - We have to set up all sprites to understand what they need to do with the points, starting with a reset of points at the beginning of the game.
 - Let's go back to the octopus. At the start of the game we always need... (by now they should know: “when the flag is clicked”). Let's move that over!
 - Back under orange Data, move “set Points to 0” under the new event header.
 - Another “when the flag is clicked” is needed for the octopus. This one will set our win conditions up.
 - To make that happen, we need Controls.
 - Just like the first row of coding on our octopus, we need a “forever” loop and an “if...then” block inside of each other.
 - This time, we're going to use a green Operator. Take the one that looks like two squares with an equal sign between them and place that inside the “if...then” block.
 - Go back to orange Data. Drag the oval that says “points” into the first square.
 - Since we have two fish to collect right now, we want it to say “if points = 2, then,” so in the second square, type 2.
 - Next we'll need to have a winning response. For the game to understand this, we need to “broadcast.” (Give them a few minutes to explore and see if they can find it).
 - Under brown Events, attach a “broadcast message1.” To make this more specific, use the dropdown to make a new message that says “you win.”

When our octopus touches the fish, we want the fish to disappear.

- Starting with Fish2, we need “When the flag is clicked” (brown Event) and “if...then” (yellow Control) wrapped by “forever.”
 - Our condition is when the octopus is touching the fish, so that would be under... (give them a minute to look) light blue Sensing: “touching mouse-pointer.”
 - In the dropdown, change it to “octopus.”
 - Our “then” reaction is to have it hide. (Again, give them a minute to see if they can find it).
 - Purple Looks, and “hide.”
 - We also need to add Data (orange): “change points by 1.”

The other fish is also going to need these same conditions, but let’s simplify our work!

- Click / hold and carefully drag that series of blocks on Fish2 over Fish3 and let go.
 - It will bounce your coding back to the right side, but (show them the coding on Fish3) there it is!
 - It copied over, but be careful not to let go over the center coding blocks – that will erase your work on the current sprite!

The crab is a little different.

- You can copy the coding from Fish2 over, but then take out the “hide.”
- We want him to broadcast a different message from Events.
 - Drag over a “broadcast you win” where “hide” had been.
 - Create a new message to say “game over.”
 - Surround all of this by a “forever loop

Back to the broadcast “you win” we created on the octopus, we need the others to receive the message.

- Starting with Fish2, under brown Events, we want “when I receive you win.”
- Again, we want the sprite to “hide” (under purple Looks).

- Drag this to Fish3 and Crab to duplicate your coding.
- For the octopus, now create another “when I receive” block with “hide” attached.
 - Make sure the received message is “game over” now.
 - Drag this to both fish.

Because the octopus and the crab will be our sign for winning or ending the game, we need them to do something special.

- For the octopus, drag over a “when I receive you win” and a “when I receive game over.”
- We’ll do the second one first.
 - If “game over,” then simply add a “hide” under this Event.
 - But for the fun part, when she wins, move her to the top right of the screen.
 - Let’s have her always move to this spot on winning and always change color.
 - We’re going to need a Motion (go to x:0 y:0) and a Looks under the “when I receive you win.”
 - Now add a “change color effect by 25.”
 - Again, we don’t want it there at the beginning, so under “change size by -20” add “set color effect to 0.”
 - Last thing for the winning octopus: if you want the color to keep changing once winning has happened, it will need a Control factor.
 - (Let them look – by now they should figure out...) A “forever” loop! Be careful to only go around the color change block.

We’re almost done! The Crab needs to do something if the game is over.

- Drag a “when I receive game over” (brown Event) over to the coding section.
- There are two options for purple “say.”
 - Change the “hello” to something silly, like “I’m too fast for you!”
- Under yellow Control, we need the crab to wait before he speaks.
 - Take the second option that has no seconds attached, so it will stay on the screen.
 - Now add at the bottom a “next costume” block.

- Control this “forever” by placing the yellow block around the costume.

To have a “you win” show up on the screen

- Go to New sprite options.
 - Only this time, click on the paintbrush.
 - Find the capital T (for text) on the editing side and click on that.
 - Click in the editing screen and type something like “You Win!”
 - You can use the edit tools at the top to make it bigger (just like we made the other sprites smaller).
 - The area with your other sprites should show an empty blue box labeled Sprite1. Click in it and the words should appear.
 - You can now drag the words in the game window down towards the bottom.
 - Go back to the Scripts tab for “you win.” We’re going to need just a few things.
 - “When the flag is clicked” (brown Event)
 - “hide” (purple Looks)
 - “When I receive you win” (brown Event)
 - “show” (purple Looks).
- Follow the same instructions to make a “Game Over” sprite.
- Then try playing the game again. You may need to go back and adjust the speed for either the crab or the octopus.

Last step!

- Click on “see project page” at the top right of the screen.
 - In the “Instructions” box, tell your players how to play. This can be something simple like: “Use arrow keys to move the octopus. Collect the fish, but avoid the crab!”
 - In the “Notes and Credits” section, you may list other examples that provided inspiration.

- Give it a name, and don't forget to click the “share” button so others can play it. That's it! Have fun playing your new game! Figures 55.1 and 55.2 show what the coding should look like.

Check out the complete animation at It's Going Swimmingly:

<https://scratch.mit.edu/projects/238290452/> .

Figure 55.1

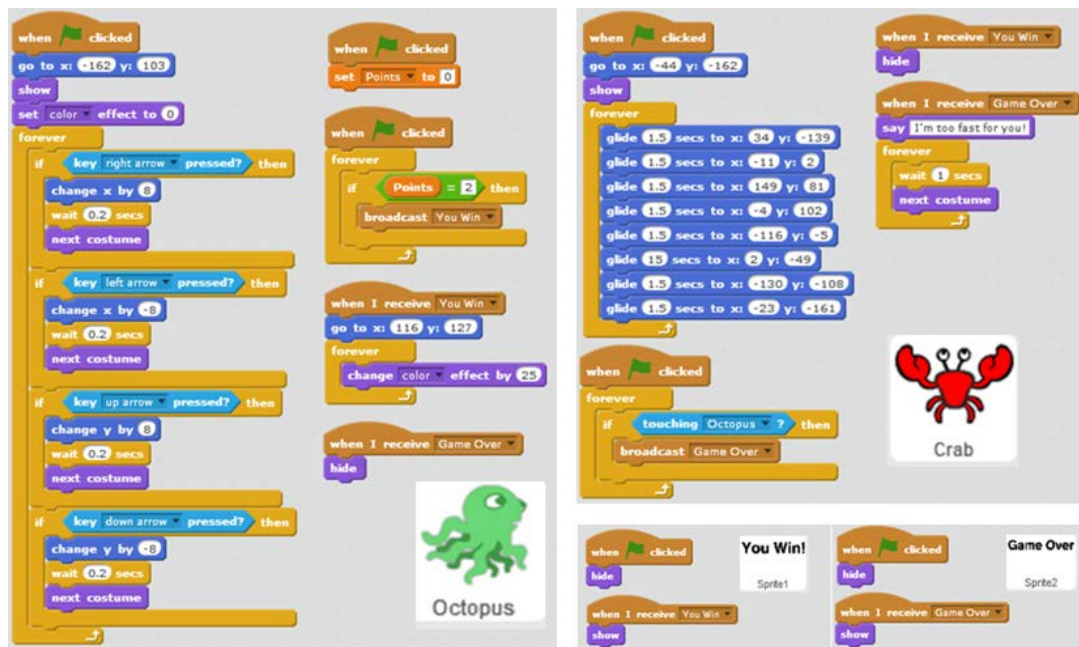
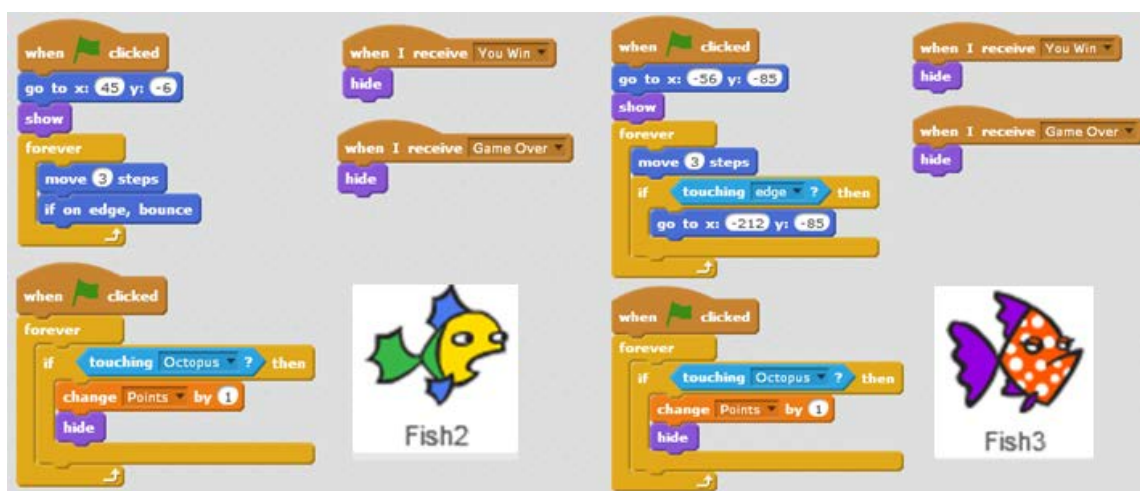


Figure 55.2



Learning Outcomes

Participants will...

- Learn to utilize the Scratch coding language to create a collecting game
- Be able to create data for point collection
- Be able to use Operator and Conditional blocks to set circumstances for actions and reactions
- Use technology to creatively express themselves
- Improve their computational thinking skills by learning how to layer and nest coding for maximum abilities

Recommended Next Projects

- “Level up” your game!
 - Add more point collecting
 - Have the sprites interact with colors
 - Add music
 - Use additional backdrops to show on “you win” or “game over” conditions
- The collectible project is a more advanced game to create and can be done on its own, but doing an earlier project like the Scratch Coding for Tweens – Creating Cartoons chapter builds a stronger foundation and aids in building confidence for those that have never coded before.
- There are numerous types of collectible games to create.
 - A next step would be to create falling “cloned” items that originate in different spots and disappear once touching the bottom of the screen.
 - Tutorials are available on YouTube to assist in navigating the steps to create such a game.
- Readers are also invited to view other projects by the author under the Scratch user name of Realityjumper1.

References

Brennan, K., Balch, C., & Chung, M. (2014). Creative Computing. *Harvard Graduate School of Education*. <http://scratched.gse.harvard.edu/guide/files/CreativeComputing20141015.pdf>

ISTE. (2018). ISTE standards for students. *International Society for Technology in Education*.
<http://www.iste.org/standards/for-students>

Scratch. (2007). <https://scratch.mit.edu/>